

Joshva J

Bachelor of Computer science and Engineering– 2024

Bengaluru, Karnataka – 580068

Task 1(Video link):

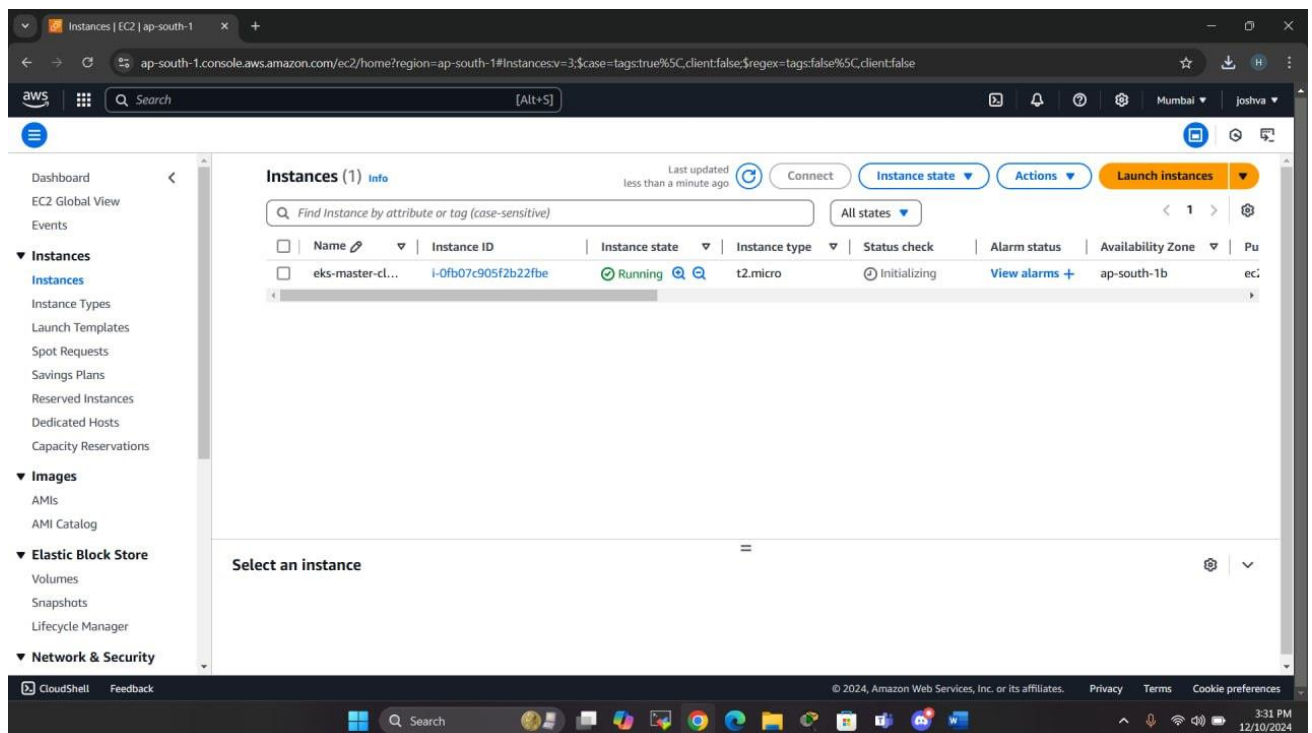
<https://drive.google.com/file/d/1upt9JdKhtANaopiIS5CfKIcqiWQ27vqN/view?usp=sharing>

DEPLOY A WEB APPLICATION IN AWS/KUBERNETES

Step 1:

Launch a New EC2 Instance (Amazon Linux - t2.micro)

- Go to the AWS Management Console and launch a new EC2 instance.
- Choose "Amazon Linux 2 AMI" as the operating system.
- Select an instance type (t2.micro is a good choice for testing purposes and free tier).
- Create new key pair and download it. Configure security groups, allowing SSH (port 22) and HTTP (port 80).
- Launch the instance and connect to it via SSH (In MobaXterm Tool).



Step 2:

Install kubectl (Kubernetes CLI)

- After connecting to the EC2 instance, run the following commands to install kubectl:

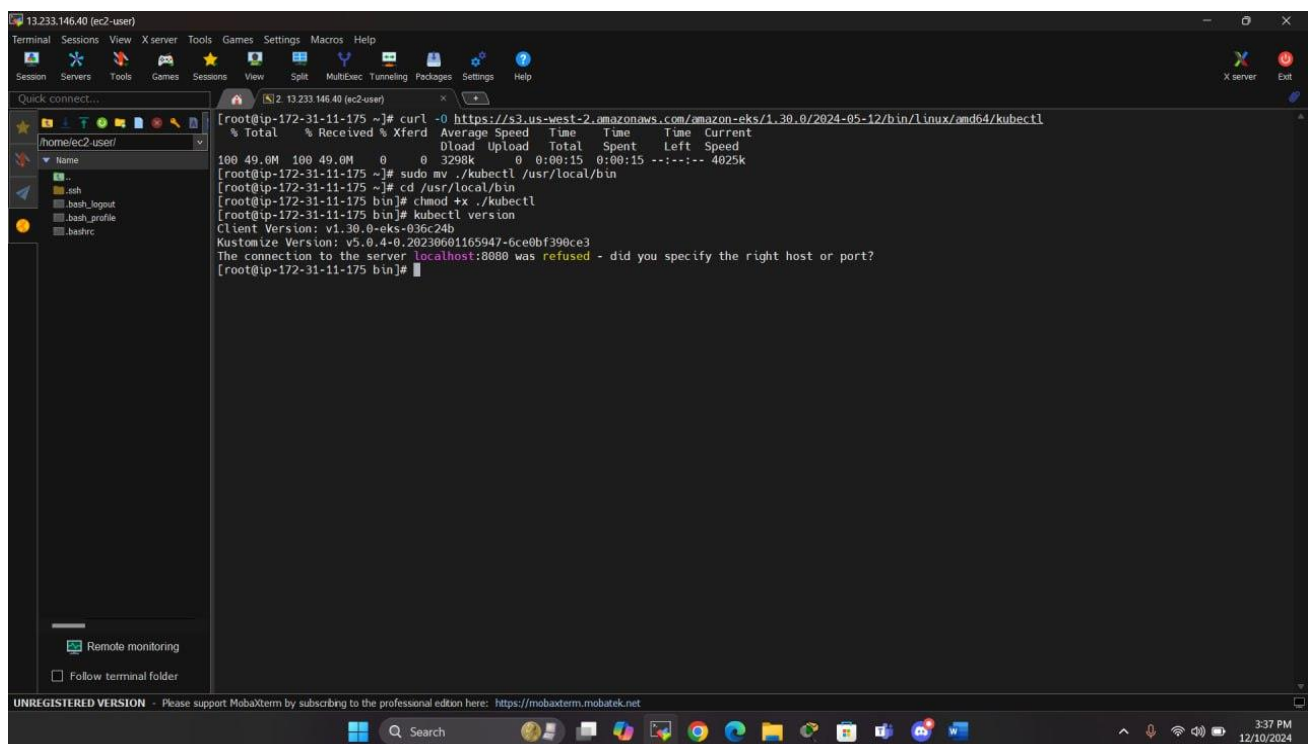
```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.0/2024-05-12/bin/linux/amd64/kubectl
```

```
sudo mv ./kubectl /usr/local/bin
```

```
chmod +x ./kubectl
```

```
kubectl version
```

- This will download kubectl, move it to /usr/local/bin for execution, apply the necessary permissions, and verify the version.

A screenshot of a terminal window running on an EC2 instance. The terminal shows the execution of several commands to install kubectl. The first command is a curl command to download kubectl from the Amazon EKS S3 bucket. The second command is a mv command to move the downloaded file to /usr/local/bin. The third command is a chmod command to set permissions. The fourth command is kubectl version, which shows the client version and a warning about the connection to the server. The terminal window has a sidebar on the left with a file explorer and a top menu bar. The bottom status bar shows the version of MobaXterm and the date and time.

```
13.233.146.40 (ec2-user)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
/home/ec2-user/
Name
ash
bash_logout
bash_profile
bashrc
Remote monitoring
Follow terminal folder
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
[root@ip-172-31-11-175 ~]# curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.0/2024-05-12/bin/linux/amd64/kubectl
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 49.6M 100 49.6M 0 0 3298k 0 0:00:15 0:00:15 --:--:-- 4025k
[root@ip-172-31-11-175 ~]# sudo mv ./kubectl /usr/local/bin
[root@ip-172-31-11-175 ~]# cd /usr/local/bin
[root@ip-172-31-11-175 bin]# chmod +x ./kubectl
[root@ip-172-31-11-175 bin]# kubectl version
Client Version: v1.30.0-eks-036c24b
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
The connection to the server localhost:8080 was refused - did you specify the right host or port?
[root@ip-172-31-11-175 bin]#
```

Step 3:

Install AWS CLI

- Install the AWS Command Line Interface (CLI) using these commands:

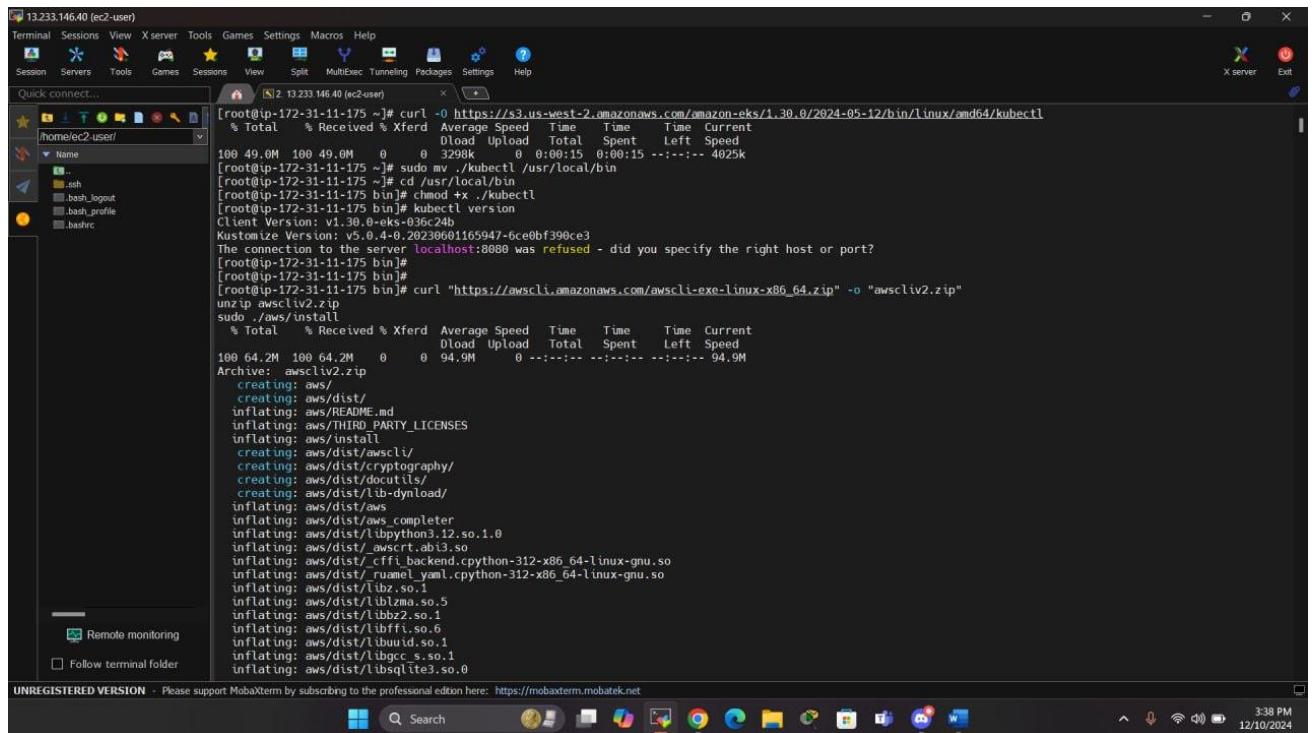
```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o  
"awscliv2.zip"
```

`unzip awscliv2.zip`

`sudo ./aws/install`

`aws --version`

- This installs the latest version of the AWS CLI and verifies the installation.



The screenshot shows a terminal window with the following commands and output:

```
[root@ip-172-31-11-175 ~]# curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.0/2024-05-12/bin/linux/amd64/kubectl
[Total % Received % Xferd Average Speed Time Time Time Current]
[0 0 0 0 0 0 0 0]
[0:00:15 0:00:15 --:--:-- 4025k]
[root@ip-172-31-11-175 ~]# sudo mv ./kubectl /usr/local/bin
[root@ip-172-31-11-175 ~]# cd /usr/local/bin
[root@ip-172-31-11-175 bin]# chmod +x ./kubectl
[root@ip-172-31-11-175 bin]# kubectl version
Client Version: v1.30.0-eks-036c24b
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
The connection to the server localhost:8080 was refused - did you specify the right host or port?
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]# curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
[Total % Received % Xferd Average Speed Time Time Time Current]
[0 0 0 0 0 0 0 0]
[0:00:15 0:00:15 --:--:-- 94.9M]
[root@ip-172-31-11-175 bin]# unzip awscliv2.zip
Archive:  awscliv2.zip
  creating: aws/
  creating: aws/dist/
  inflating: aws/README.md
  inflating: aws/THIRD_PARTY_LICENSES
  inflating: aws/install
  creating: aws/dist/awscli/
  creating: aws/dist/cryptography/
  creating: aws/dist/docutils/
  creating: aws/dist/lib-dynload/
  inflating: aws/dist/aws
  inflating: aws/dist/aws_completer
  inflating: aws/dist/libpython3.12.so.1.0
  inflating: aws/dist/_awsrt.abi3.so
  inflating: aws/dist/_cffi_backend.cpython-312-x86_64-linux-gnu.so
  inflating: aws/dist/_ruamel.yaml.cpython-312-x86_64-linux-gnu.so
  inflating: aws/dist/libz.so.1
  inflating: aws/dist/liblzma.so.5
  inflating: aws/dist/libbz2.so.1
  inflating: aws/dist/libffi.so.6
  inflating: aws/dist/libuuid.so.1
  inflating: aws/dist/libgcc_s.so.1
  inflating: aws/dist/libsqlite3.so.0
[root@ip-172-31-11-175 bin]# sudo ./aws/install
[Total % Received % Xferd Average Speed Time Time Time Current]
[0 0 0 0 0 0 0 0]
[0:00:15 0:00:15 --:--:-- 94.9M]
[root@ip-172-31-11-175 bin]#
```

Step 4:

Install eksctl (EKS Cluster Management Tool)

- To simplify EKS cluster creation, install eksctl:

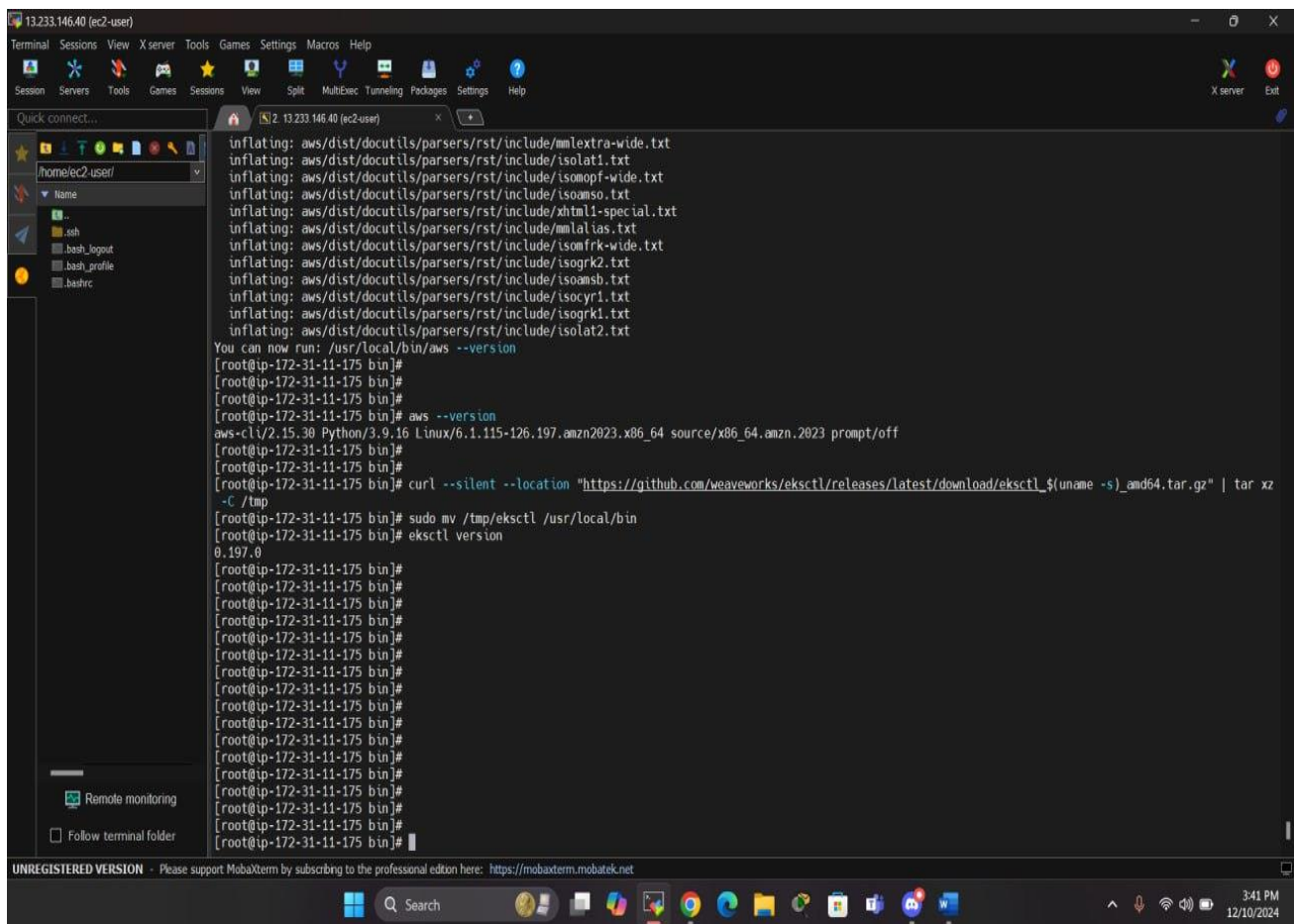
`curl --silent --location`

`"https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp`

`sudo mv /tmp/eksctl /usr/local/bin`

`eksctl version`

- This command downloads and installs eksctl, a command-line tool for managing EKS clusters.



```
13.233.146.40 (ec2-user)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
/home/ec2-user/
Name
ssh
bash_logout
bash_profile
bashrc
Remote monitoring
Follow terminal folder
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

inflating: aws/dist/docutils/parsers/rst/include/mmlextra-wide.txt
inflating: aws/dist/docutils/parsers/rst/include/isolat1.txt
inflating: aws/dist/docutils/parsers/rst/include/isoamsp-wide.txt
inflating: aws/dist/docutils/parsers/rst/include/isoamso.txt
inflating: aws/dist/docutils/parsers/rst/include/xhtml11-special.txt
inflating: aws/dist/docutils/parsers/rst/include/mmlalias.txt
inflating: aws/dist/docutils/parsers/rst/include/isoamfrk-wide.txt
inflating: aws/dist/docutils/parsers/rst/include/isogr2.txt
inflating: aws/dist/docutils/parsers/rst/include/isoamsb.txt
inflating: aws/dist/docutils/parsers/rst/include/isocyr1.txt
inflating: aws/dist/docutils/parsers/rst/include/isogr1.txt
inflating: aws/dist/docutils/parsers/rst/include/isolat2.txt
You can now run: /usr/local/bin/aws --version
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]# aws --version
aws-cli/2.15.30 Python/3.9.16 Linux/6.1.115-126.197.amzn2023.x86_64 source/x86_64.amzn.2023 prompt/off
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]# curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_${uname -s}_amd64.tar.gz" | tar xz
-C /tmp
[root@ip-172-31-11-175 bin]# sudo mv /tmp/eksctl /usr/local/bin
[root@ip-172-31-11-175 bin]# eksctl version
0.197.0
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
```

Step 5:

Create a New IAM Role

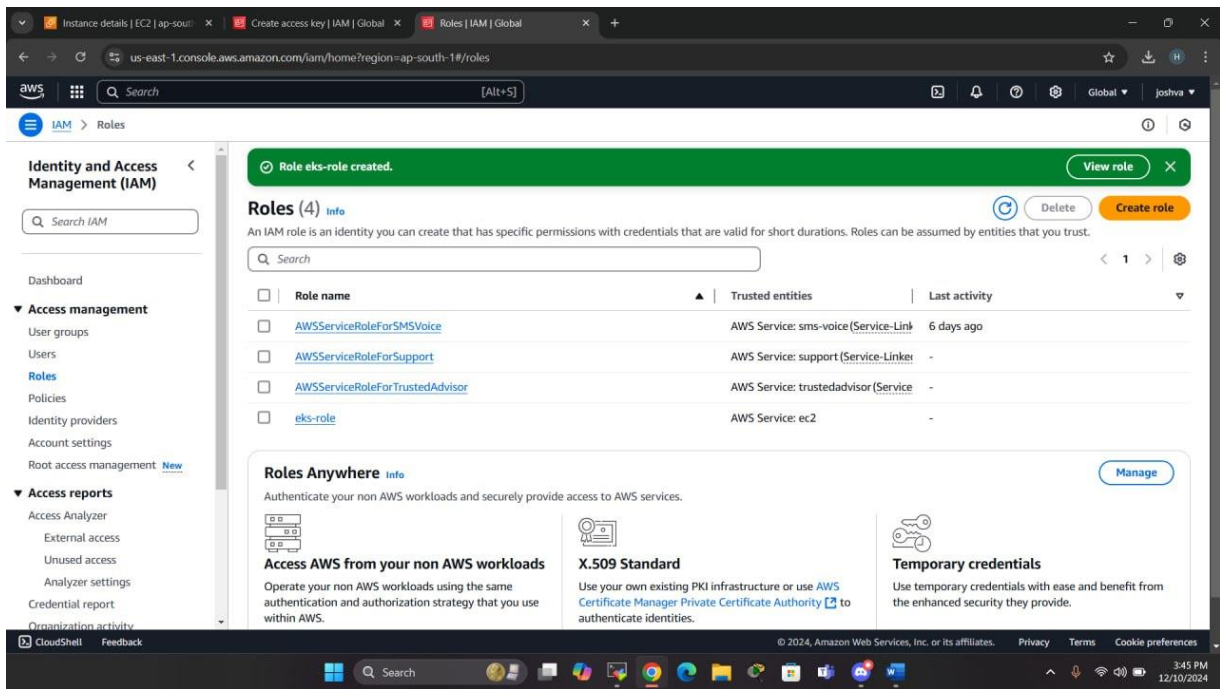
- Go to the IAM section in the AWS Console and create a new role with the following permissions:

- IAM FullAccess – We need create all these roles individually

(or)

Otherwise we give one role is “AdministratorAccess”

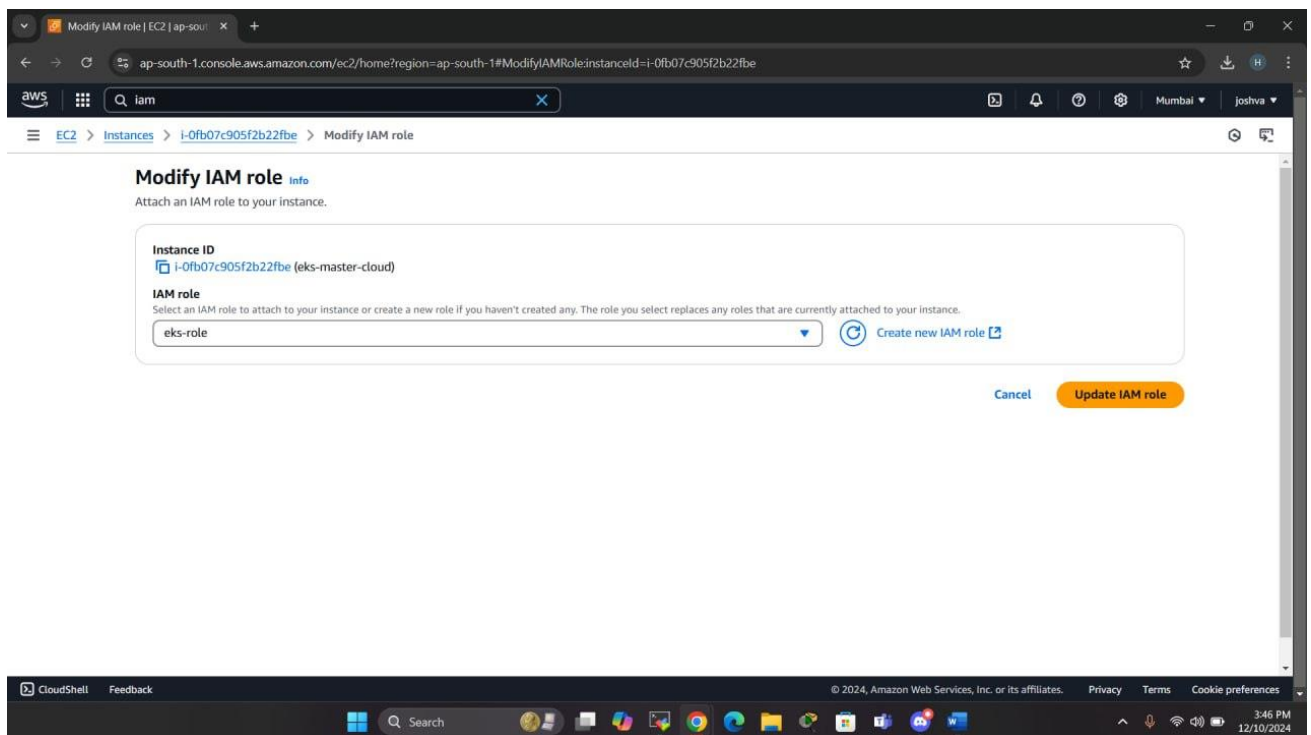
(VPC FullAccess,EC2 FullAccess, CloudFormation FullAccess).



Step 6:

Attach IAM Role to the EC2 Instance

- Attach the created role to the EC2 instance used as the management host. This allows the instance to interact with other AWS services.



Step 7:

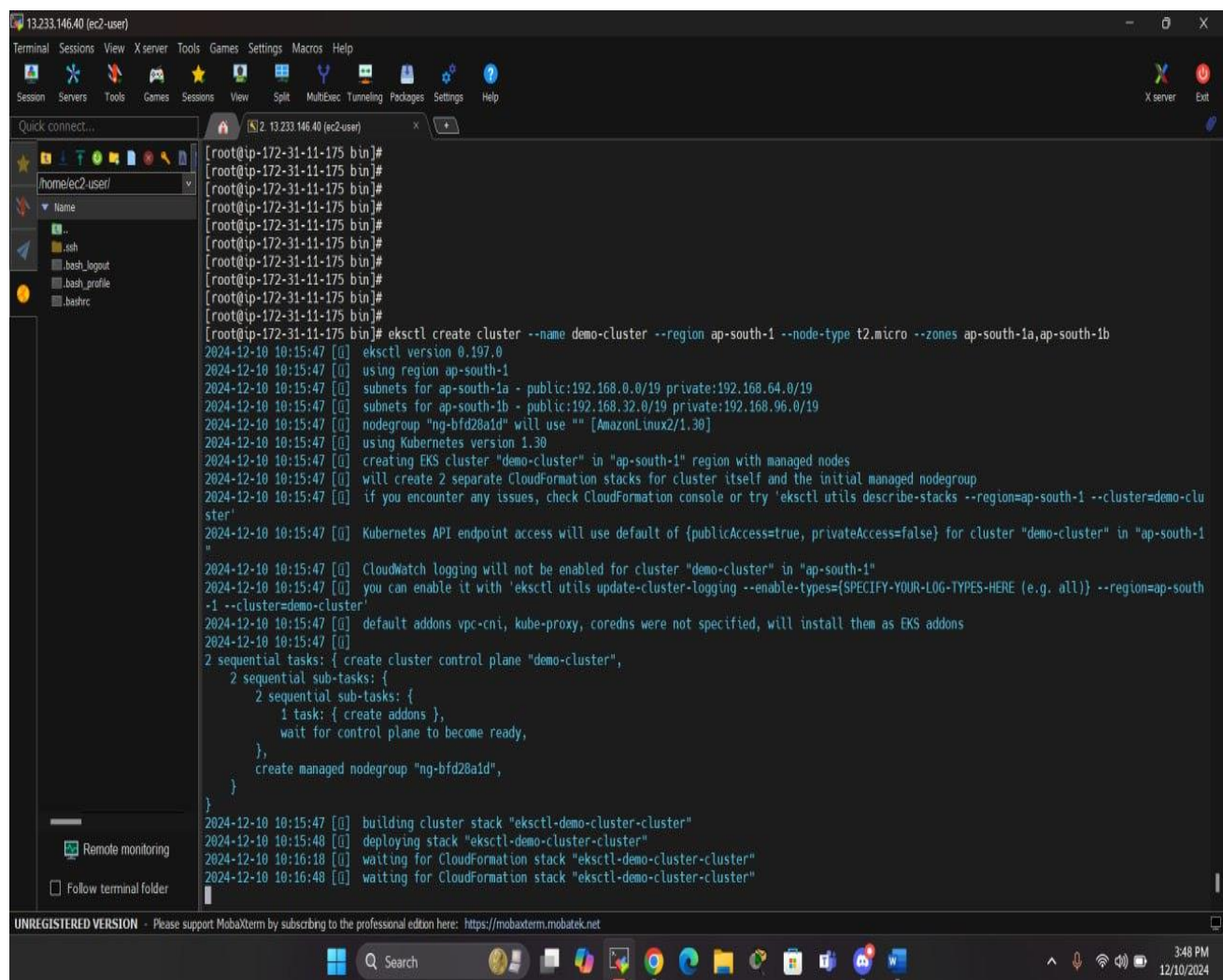
Create EKS Cluster using eksctl

- To create the EKS cluster, use the eksctl command.

Mumbai (ap-south-1):

```
eksctl create cluster --name demo-cluster --region ap-south-1 --node-type t2.micro --zones ap-south-1a,ap-south-1b
```

- This will create a new EKS cluster in the specified region, using the specified instance type and zones.



```
13.233.146.40 (ec2-user)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
/home/ec2-user/
Name
.
ssh
.bash_logout
.bash_profile
.bashrc
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]# eksctl create cluster --name demo-cluster --region ap-south-1 --node-type t2.micro --zones ap-south-1a,ap-south-1b
2024-12-10 10:15:47 [i] eksctl version 0.197.0
2024-12-10 10:15:47 [i] using region ap-south-1
2024-12-10 10:15:47 [i] subnets for ap-south-1a - public:192.168.0.0/19 private:192.168.64.0/19
2024-12-10 10:15:47 [i] subnets for ap-south-1b - public:192.168.32.0/19 private:192.168.96.0/19
2024-12-10 10:15:47 [i] nodegroup "ng-bfd28a1d" will use "" [AmazonLinux2/1.30]
2024-12-10 10:15:47 [i] using Kubernetes version 1.30
2024-12-10 10:15:47 [i] creating EKS cluster "demo-cluster" in "ap-south-1" region with managed nodes
2024-12-10 10:15:47 [i] will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2024-12-10 10:15:47 [i] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=ap-south-1 --cluster=demo-cluster'
2024-12-10 10:15:47 [i] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "demo-cluster" in "ap-south-1"
2024-12-10 10:15:47 [i] CloudWatch logging will not be enabled for cluster "demo-cluster" in "ap-south-1"
2024-12-10 10:15:47 [i] you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=ap-south-1 --cluster=demo-cluster'
2024-12-10 10:15:47 [i] default addons vpc-cni, kube-proxy, coredns were not specified, will install them as EKS addons
2024-12-10 10:15:47 [i]
2 sequential tasks: { create cluster control plane "demo-cluster",
  2 sequential sub-tasks: {
    2 sequential sub-tasks: {
      1 task: { create addons },
      wait for control plane to become ready,
    },
    create managed nodegroup "ng-bfd28a1d",
  }
}
2024-12-10 10:15:47 [i] building cluster stack "eksctl-demo-cluster-cluster"
2024-12-10 10:15:48 [i] deploying stack "eksctl-demo-cluster-cluster"
2024-12-10 10:16:18 [i] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-12-10 10:16:48 [i] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
```

The screenshot shows a MobaXterm terminal window with a dark theme. The title bar indicates the connection is to 13.233.146.40 (ec2-user). The terminal output shows the following sequence of events:

- Building managed nodegroup stack "eksctl-demo-cluster-nodegroup-ng-bfd28aid"
- Deploying stack "eksctl-demo-cluster-nodegroup-ng-bfd28aid"
- Waiting for CloudFormation stack "eksctl-demo-cluster-nodegroup-ng-bfd28aid"
- Waiting for the control plane to become ready
- Saved kubeconfig as "/root/.kube/config"
- no tasks
- all EKS cluster resources for "demo-cluster" have been created
- nodegroup "ng-bfd28aid" has 2 node(s)
- node "ip-192-168-10-199.ap-south-1.compute.internal" is ready
- node "ip-192-168-63-45.ap-south-1.compute.internal" is ready
- waiting for at least 2 node(s) to become ready in "ng-bfd28aid"
- nodegroup "ng-bfd28aid" has 2 node(s)
- node "ip-192-168-10-199.ap-south-1.compute.internal" is ready
- node "ip-192-168-63-45.ap-south-1.compute.internal" is ready
- created 1 managed nodegroup(s) in cluster "demo-cluster"
- kubectl command should work with "/root/.kube/config", try 'kubectl get nodes'
- EKS cluster "demo-cluster" in "ap-south-1" region is ready

The user then runs the following commands:

```
[root@ip-172-31-11-175 bin]#  
[root@ip-172-31-11-175 bin]#  
[root@ip-172-31-11-175 bin]# kubectl create deployment demo-httpd2 --image=ss1927/httpd2 --replicas=1 --port=80  
deployment.apps/demo-httpd2 created  
[root@ip-172-31-11-175 bin]#  
[root@ip-172-31-11-175 bin]# kubectl run webapp --image=ss1927/httpd2  
pod/webapp created  
[root@ip-172-31-11-175 bin]# kubectl get nodes:  
error: the server doesn't have a resource type "nodes:"  
[root@ip-172-31-11-175 bin]# kubectl get nodes  
[root@ip-172-31-11-175 bin]# kubectl get pods  
NAME READY STATUS RESTARTS AGE  
demo-httpd2-5f87f67bd9-vcpbv 1/1 Running 0 105s  
webapp 1/1 Running 0 61s  
[root@ip-172-31-11-175 bin]#  
[root@ip-172-31-11-175 bin]#  
[root@ip-172-31-11-175 bin]#
```

The terminal window also shows a sidebar with file explorer and session management options. At the bottom, there is a notice about the unregistered version of MobaXterm and a Windows taskbar with the time 4:03 PM on 12/10/2024.

Step 8:

Deploy Nginx Pods on Kubernetes

- Create a Deployment for Nginx: To deploy an given demo-web-httpd application with 1 replicas:

```
kubectl create deployment demo-web-httpd --image=ss1927/httpd --replicas=1 --port=80
```

- Check the Status:

List all resources - `kubectl get all`

List the running pods - `kubectl get pods`

- After all this, To run the webapp with help this command:

```
Kubectl run webapp --image=ss1927/httpd
```

The screenshot shows a MobaXterm terminal window with a file explorer on the left. The terminal output shows the following sequence of commands and results:

```
2024-12-10 10:24:50 [i] building managed nodegroup stack "eksctl-demo-cluster-nodegroup-ng-bfd28a1d"
2024-12-10 10:24:50 [i] deploying stack "eksctl-demo-cluster-nodegroup-ng-bfd28a1d"
2024-12-10 10:24:50 [i] waiting for CloudFormation stack "eksctl-demo-cluster-nodegroup-ng-bfd28a1d"
2024-12-10 10:25:52 [i] waiting for CloudFormation stack "eksctl-demo-cluster-nodegroup-ng-bfd28a1d"
2024-12-10 10:26:50 [i] waiting for CloudFormation stack "eksctl-demo-cluster-nodegroup-ng-bfd28a1d"
2024-12-10 10:28:18 [i] waiting for CloudFormation stack "eksctl-demo-cluster-nodegroup-ng-bfd28a1d"
2024-12-10 10:28:18 [i] waiting for the control plane to become ready
2024-12-10 10:28:19 [i] saved kubeconfig as "/root/.kube/config"
2024-12-10 10:28:19 [i] no tasks
2024-12-10 10:28:19 [i] all EKS cluster resources for "demo-cluster" have been created
2024-12-10 10:28:19 [i] nodegroup "ng-bfd28a1d" has 2 node(s)
2024-12-10 10:28:19 [i] node "ip-192-168-10-199.ap-south-1.compute.internal" is ready
2024-12-10 10:28:19 [i] node "ip-192-168-63-45.ap-south-1.compute.internal" is ready
2024-12-10 10:28:19 [i] waiting for at least 2 node(s) to become ready in "ng-bfd28a1d"
2024-12-10 10:28:19 [i] nodegroup "ng-bfd28a1d" has 2 node(s)
2024-12-10 10:28:19 [i] node "ip-192-168-10-199.ap-south-1.compute.internal" is ready
2024-12-10 10:28:19 [i] node "ip-192-168-63-45.ap-south-1.compute.internal" is ready
2024-12-10 10:28:19 [i] created 1 managed nodegroup(s) in cluster "demo-cluster"
2024-12-10 10:28:20 [i] kubectl command should work with "/root/.kube/config", try 'kubectl get nodes'
2024-12-10 10:28:20 [i] EKS cluster "demo-cluster" in "ap-south-1" region is ready

[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]# kubectl create deployment demo-httpd2 --image=ss1927/httpd2 --replicas=1 --port=80
deployment.apps/demo-httpd2 created
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]# kubectl run webapp --image=ss1927/httpd2
pod/webapp created
[root@ip-172-31-11-175 bin]# kubectl get nodes:
error: the server doesn't have a resource type "nodes:"
[root@ip-172-31-11-175 bin]# kubectl get nodes
NAME                                STATUS    ROLES    AGE    VERSION
ip-192-168-10-199.ap-south-1.compute.internal Ready    <none>   4m29s  v1.30.6-eks-94953ac
ip-192-168-63-45.ap-south-1.compute.internal Ready    <none>   4m28s  v1.30.6-eks-94953ac
[root@ip-172-31-11-175 bin]# kubectl get pods
NAME                                READY    STATUS    RESTARTS  AGE
demo-httpd2-5f87f67bd9-vcpbv        1/1      Running   0          105s
webapp                               1/1      Running   0          61s
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
```

Step 9:

Expose the Deployment as a Service

- To expose the Nginx deployment via a LoadBalancer, follow these steps:

Expose the Deployment: `kubectl expose deployment demo--web-httpd --port=80 --type=LoadBalancer`

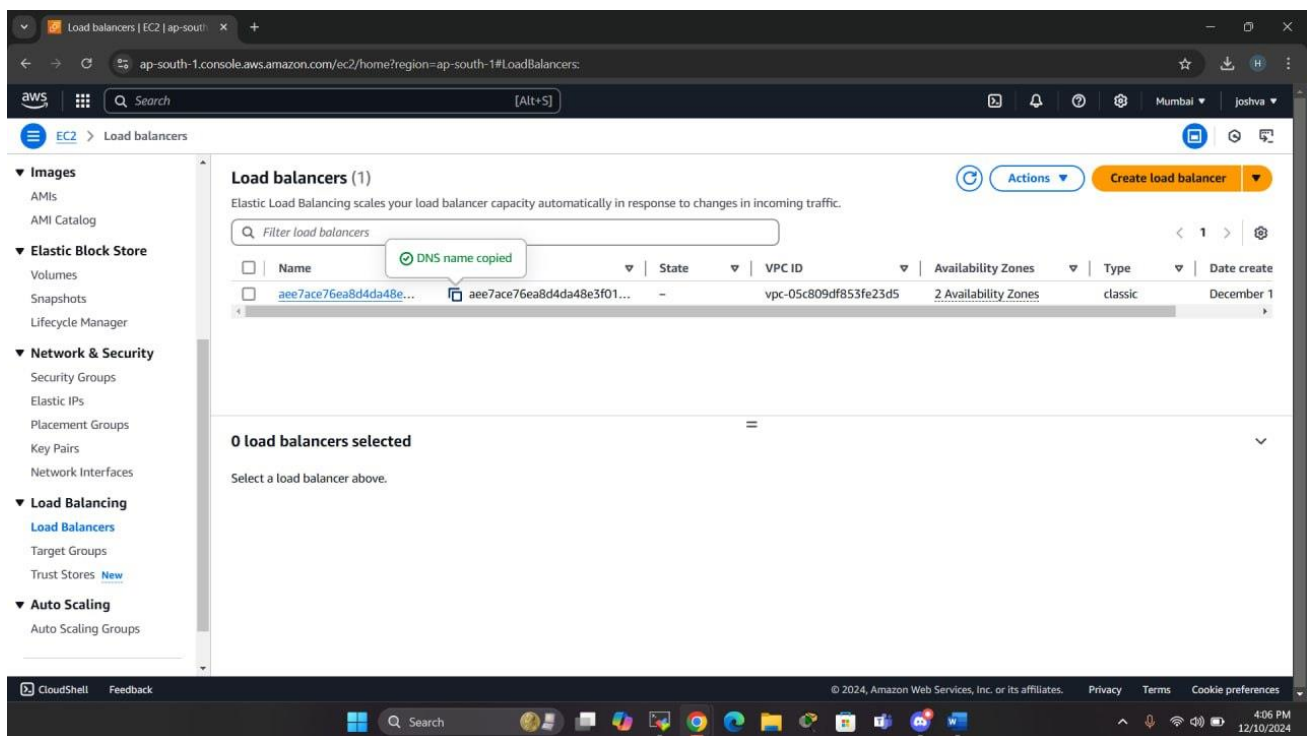
Check the Service: To see the service details and external IP

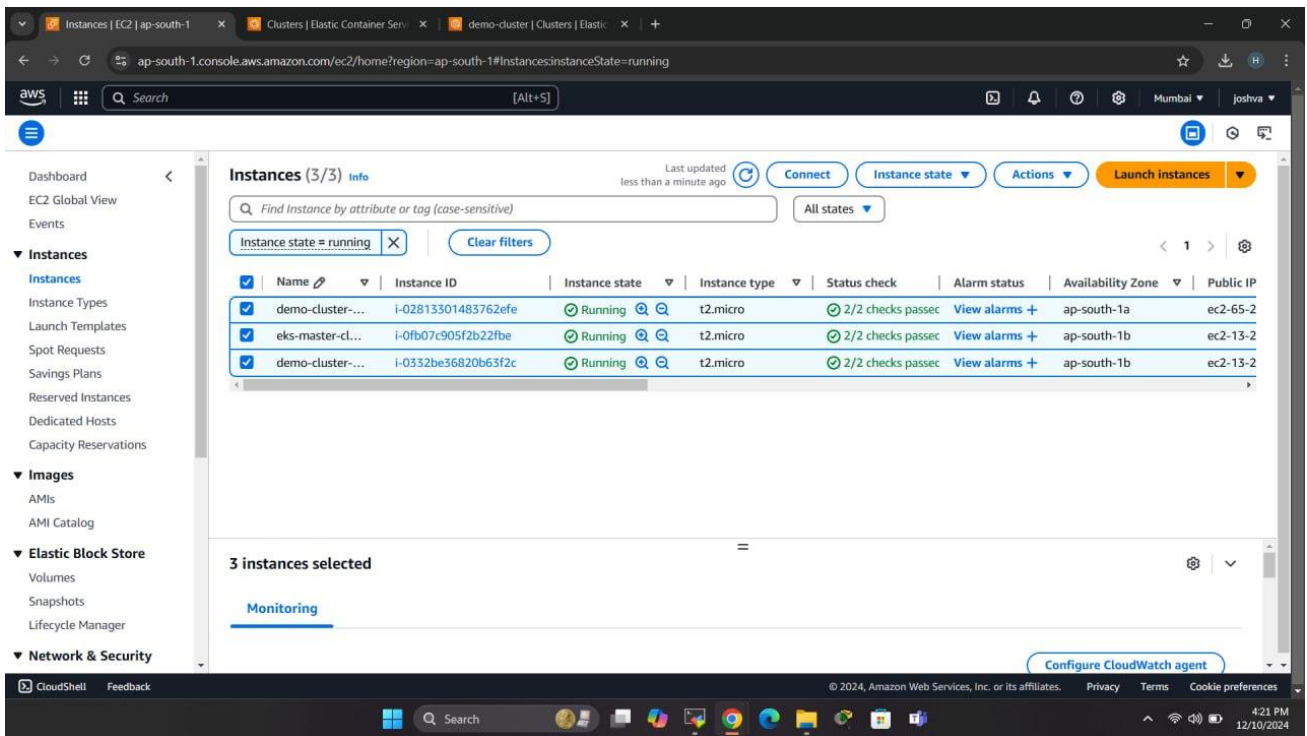
(LoadBalancer IP)

`kubectl get services -o wide`


```
13.233.146.40 (ec2-user)
Terminal Sessions View X server Tools Games Settings Macros Help
Quick connect...
/home/ec2-user/
Name
ssh
bash_logout
bash_profile
bashrc
Remote monitoring
Follow terminal folder

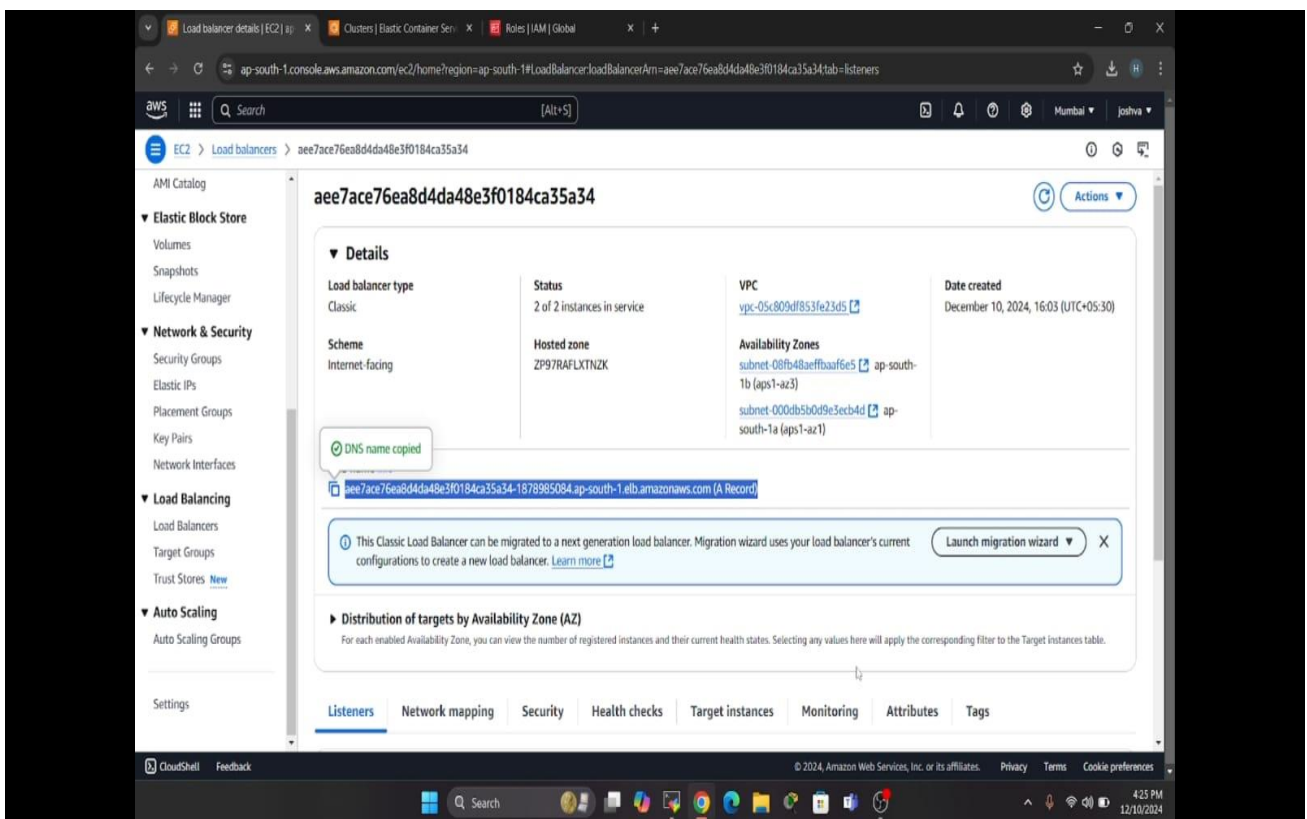
2024-12-10 10:26:50 [0] waiting for CloudFormation stack "eksctl-demo-cluster-nodegroup-ng-bfd28aid"
2024-12-10 10:28:18 [0] waiting for CloudFormation stack "eksctl-demo-cluster-nodegroup-ng-bfd28aid"
2024-12-10 10:28:18 [0] waiting for the control plane to become ready
2024-12-10 10:28:19 [✓] saved kubeconfig as "/root/.kube/config"
2024-12-10 10:28:19 [0] no tasks
2024-12-10 10:28:19 [✓] all EKS cluster resources for "demo-cluster" have been created
2024-12-10 10:28:19 [0] nodegroup "ng-bfd28aid" has 2 node(s)
2024-12-10 10:28:19 [0] node "ip-192-168-10-199.ap-south-1.compute.internal" is ready
2024-12-10 10:28:19 [0] node "ip-192-168-63-45.ap-south-1.compute.internal" is ready
2024-12-10 10:28:19 [0] waiting for at least 2 node(s) to become ready in "ng-bfd28aid"
2024-12-10 10:28:19 [0] nodegroup "ng-bfd28aid" has 2 node(s)
2024-12-10 10:28:19 [0] node "ip-192-168-10-199.ap-south-1.compute.internal" is ready
2024-12-10 10:28:19 [0] node "ip-192-168-63-45.ap-south-1.compute.internal" is ready
2024-12-10 10:28:19 [✓] created 1 managed nodegroup(s) in cluster "demo-cluster"
2024-12-10 10:28:20 [0] kubect command should work with "/root/.kube/config", try 'kubectl get nodes'
2024-12-10 10:28:20 [✓] EKS cluster "demo-cluster" in "ap-south-1" region is ready
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]# kubectl create deployment demo-httpd2 --image=ss1927/httpd2 --replicas=1 --port=80
deployment.apps/demo-httpd2 created
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]# kubectl run webapp --image=ss1927/httpd2
pod/webapp created
[root@ip-172-31-11-175 bin]# kubectl get nodes:
error: the server doesn't have a resource type "nodes:"
[root@ip-172-31-11-175 bin]# kubectl get nodes
NAME                                STATUS    ROLES    AGE   VERSION
ip-192-168-10-199.ap-south-1.compute.internal Ready    <none>   4m29s v1.30.6-eks-94953ac
ip-192-168-63-45.ap-south-1.compute.internal Ready    <none>   4m28s v1.30.6-eks-94953ac
[root@ip-172-31-11-175 bin]# kubectl get pods
NAME                                READY    STATUS    RESTARTS   AGE
demo-httpd2-5f87f67bd9-vcpbv        1/1      Running   0           105s
webapp                               1/1      Running   0           61s
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]# kubectl expose deployment demo-httpd2 --port=80 --type=LoadBalancer
Error from server (NotFound): deployments.apps "demo-httpd2" not found
[root@ip-172-31-11-175 bin]# kubectl expose deployment demo-httpd2 --port=80 --type=LoadBalancer
service/demo-httpd2 exposed
[root@ip-172-31-11-175 bin]#
[root@ip-172-31-11-175 bin]#
```

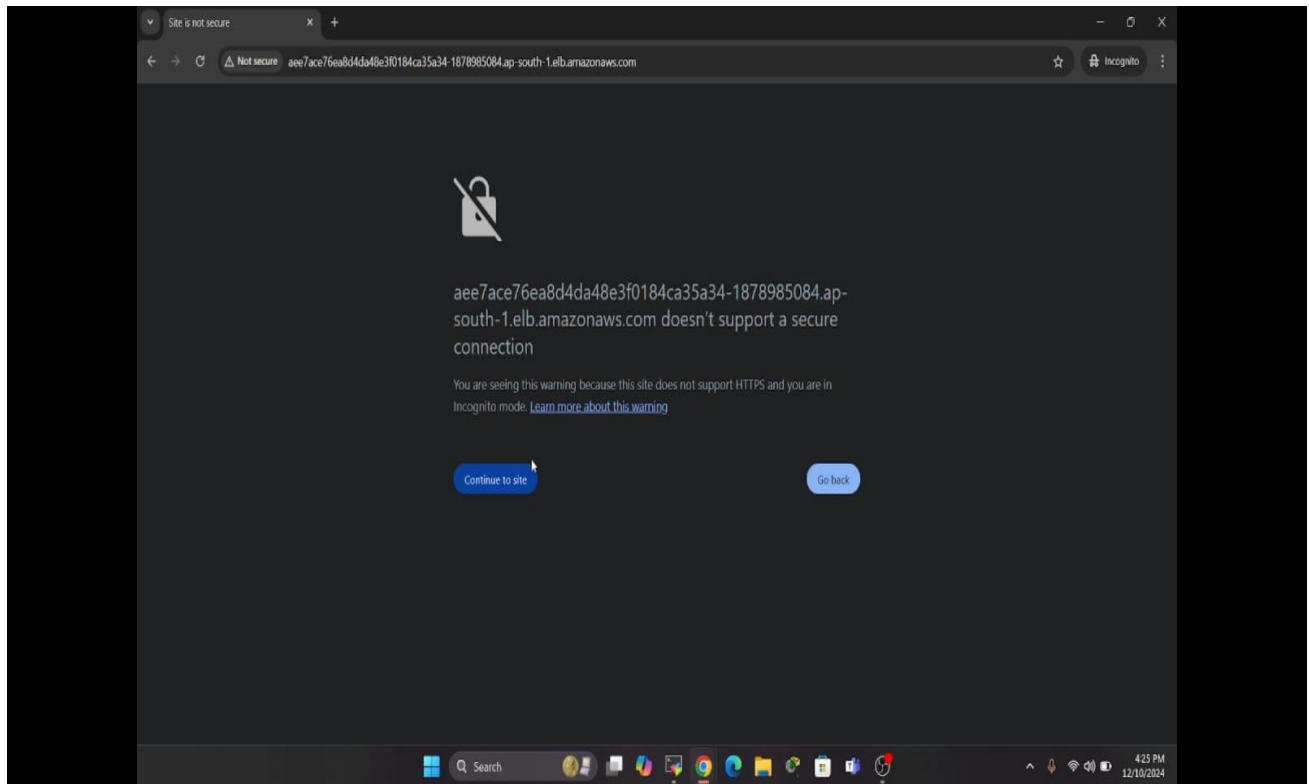
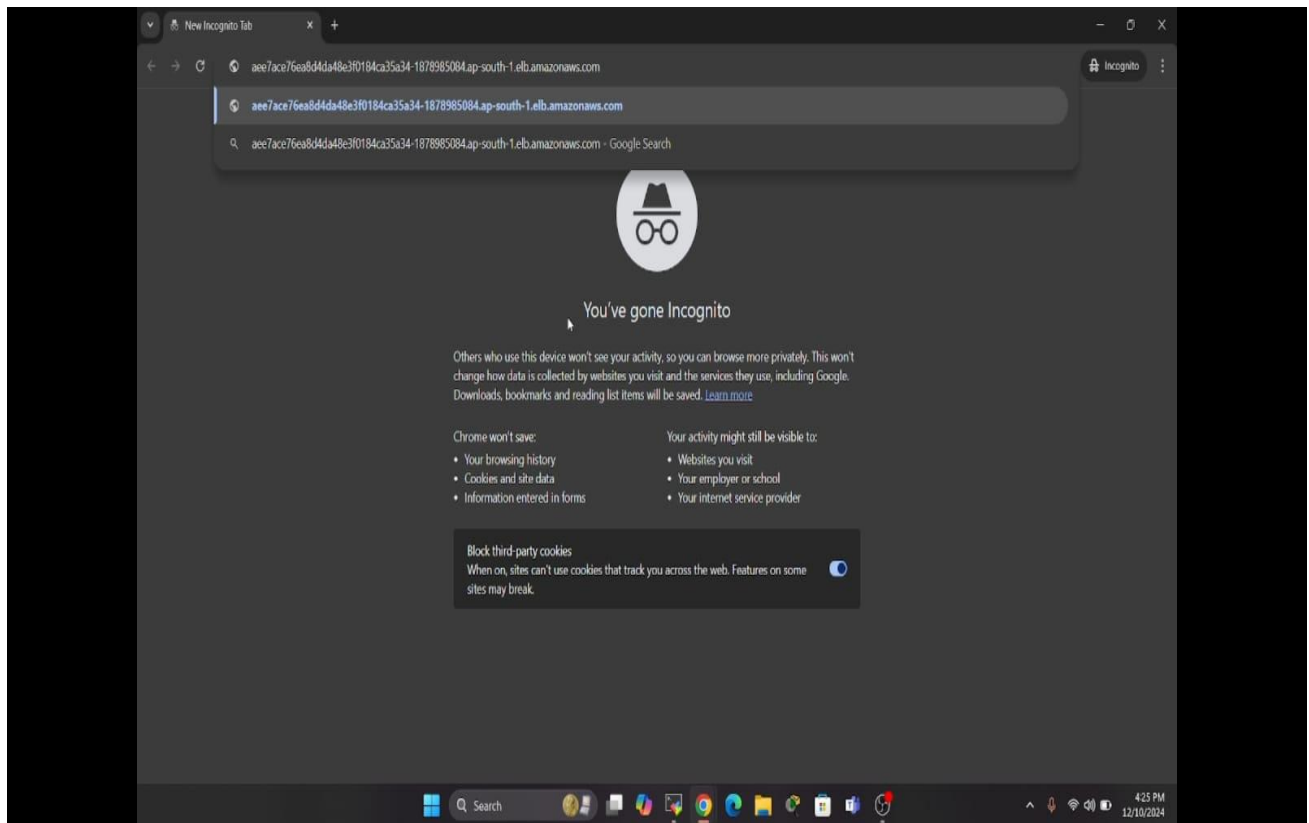


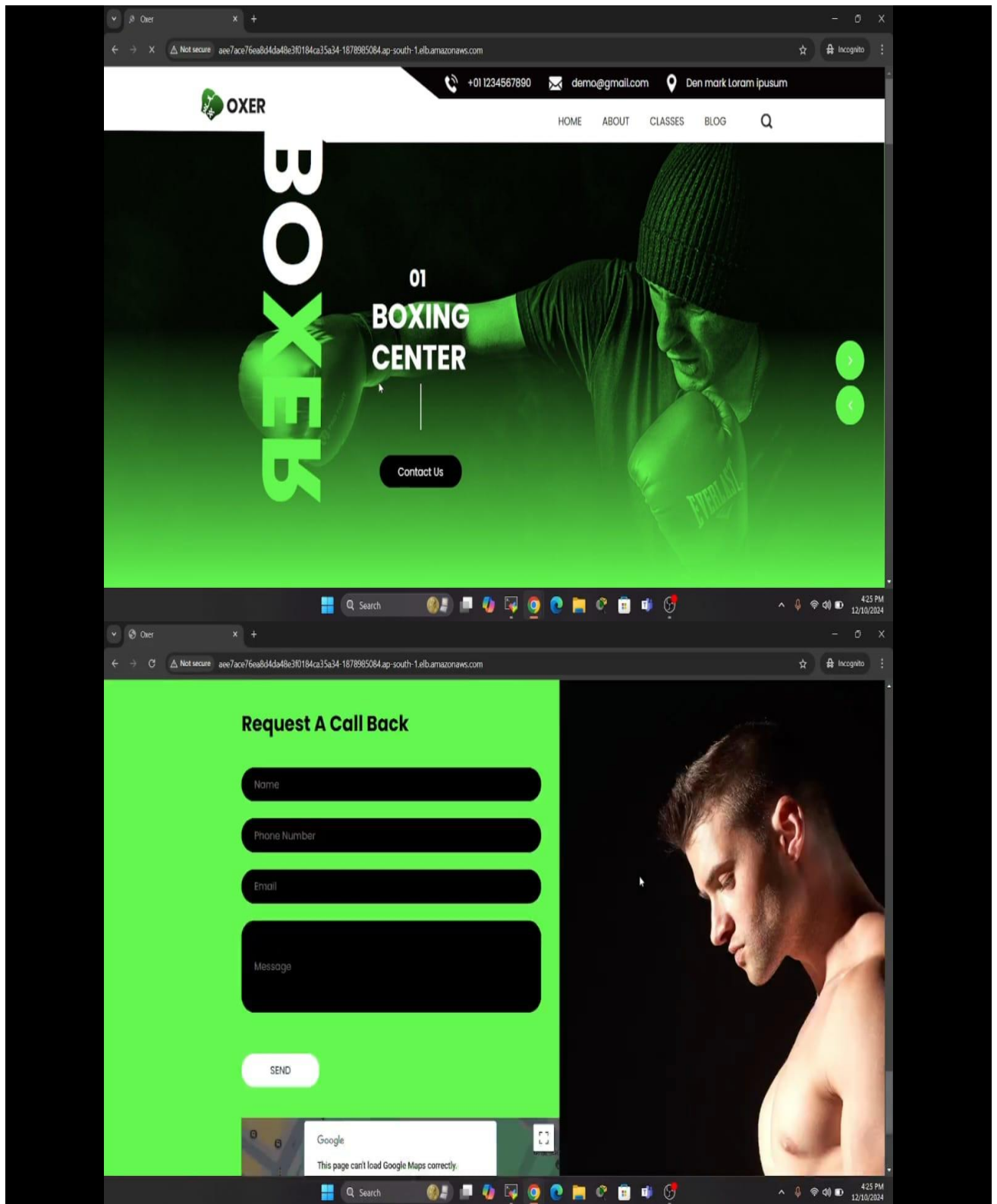


Step 10:

Output







Thank You